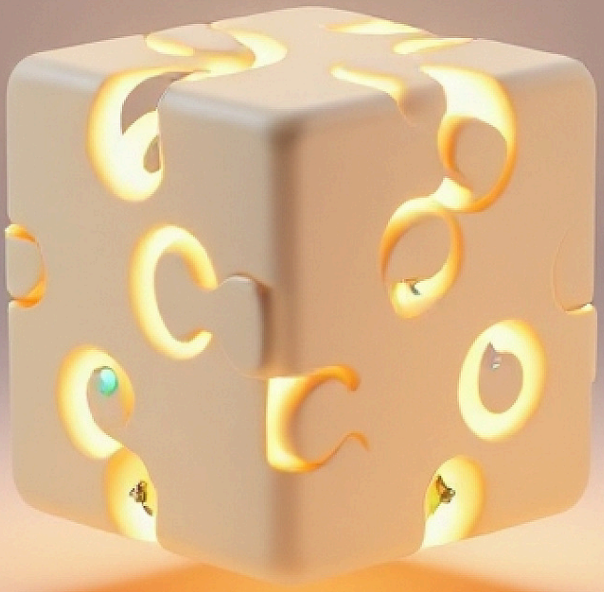
Nesne yönelimli programlama nedir?

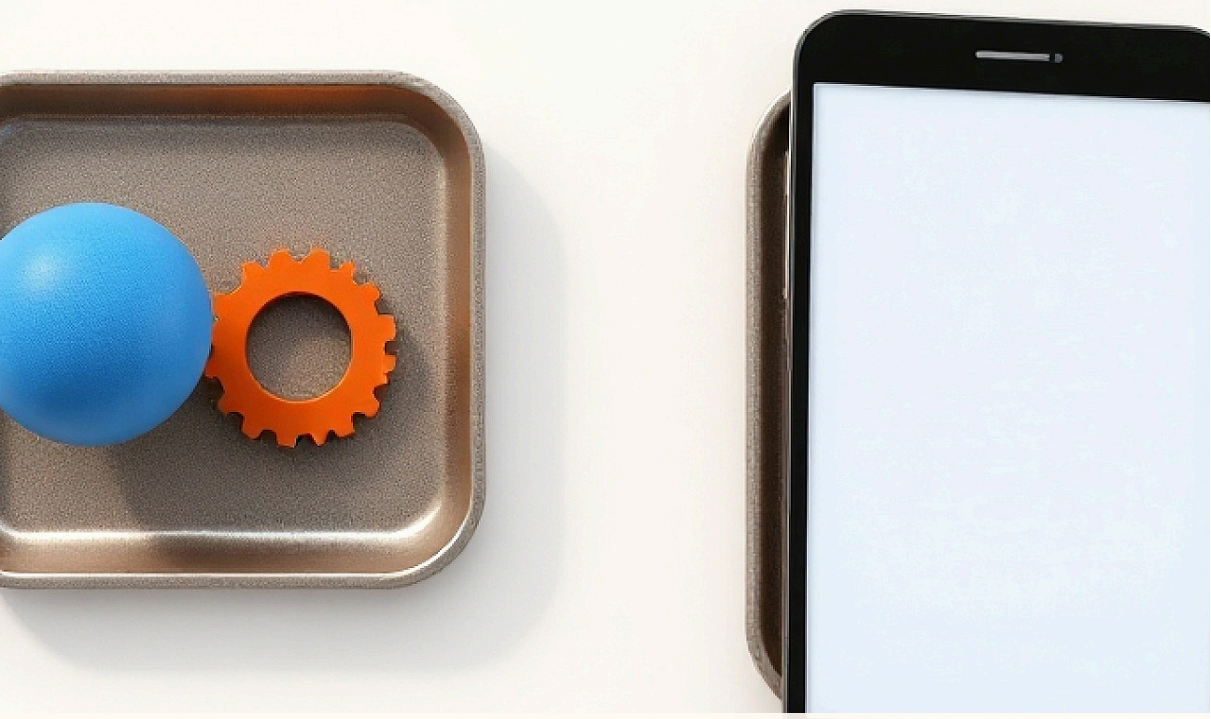


www.yzokul.com.tr

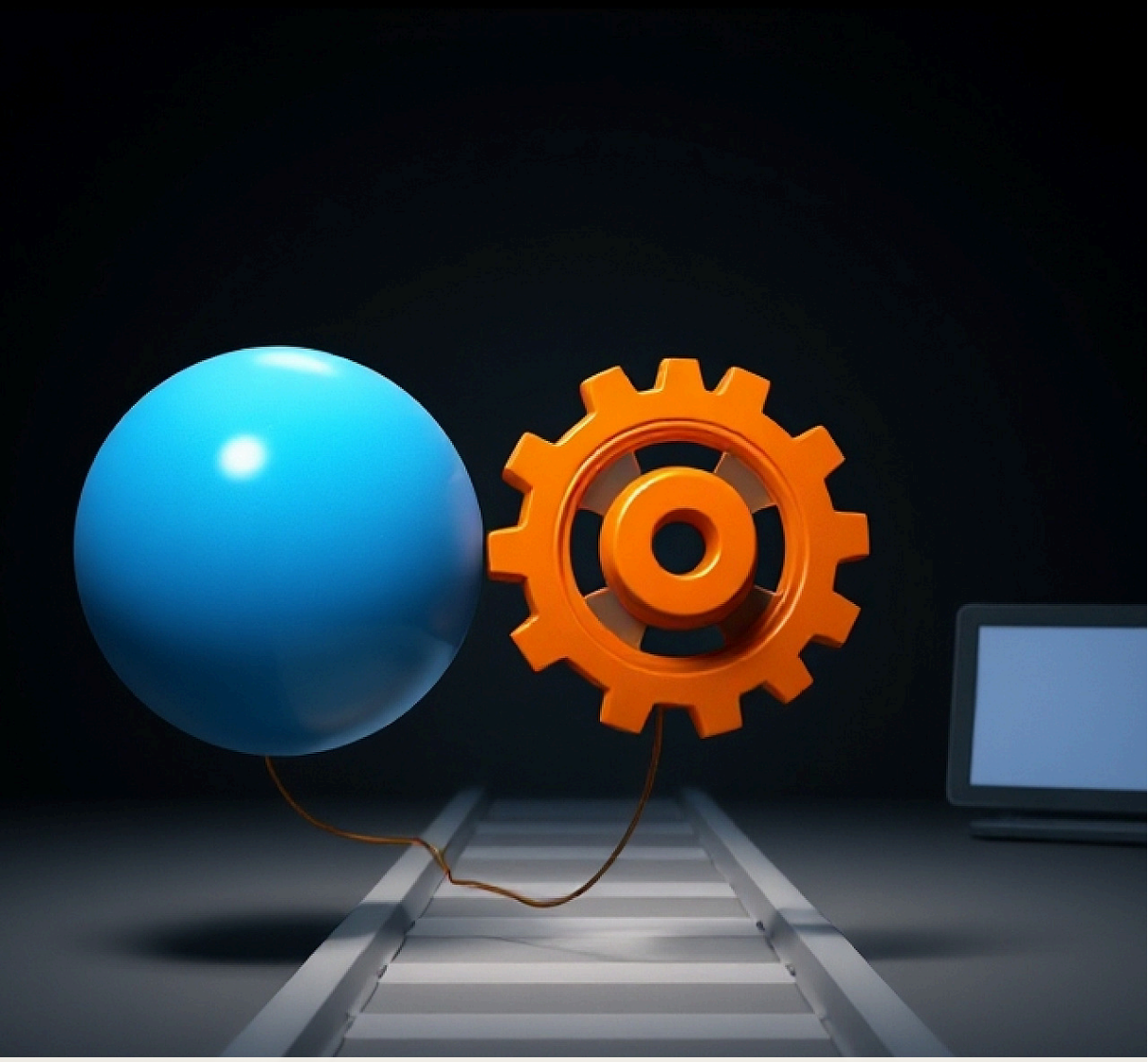
Nesne yönelimli programlama nedir?



Nesne yönelimli programlama terimini duymuş ve ne anlama geldiğini merak etmiş olabilirsin. Kısa yanıt: bu, bir programı veriler ve o veriler üzerindeki eylemler nesneler etrafında gruplanacak şekilde düzenleme biçimidir. Bu açıklamanın geri kalanını izlemek için önceden hiçbir bilgiye ihtiyacın yok. Her şey sıfırdan, her seferinde bir fikirle inşa edilir.



Her program bir miktar veri tutar ve bu veri üzerinde bazı eylemler gerçekleştirir. Veri olarak bir sayı saklayan ve ona bir ekleyen bir eylemi olan basit bir program düşün. Veri bir yerde, eylem başka bir yerde durur. Onları ayrı tutmak küçük programlarda işe yarar, ama programlar büyüdükçe zorlaşır. Bu ayrılık, nesnelerin neden yardımcı olduğunu anlamanın başlangıç noktasıdır.



Bir programda veri ve onu kullanan eylemler birbirinden uzak tutulduğunda, hangi eylemin hangi veriye ait olduğunu görmek zorlaşır. Çok sayıda eylemin ve çok sayıda veri parçan olabilir ve hiçbir şey bunların nasıl bağlandığını göstermez. Bir yerdeki değişiklik uzaktaki bir şeyi bozabilir. Bu kopukluk programları okumayı kafa karıştırıcı, değiştirmeyi de riskli hale getirir; özellikle de aynı kod üzerinde birçok kişi çalıştığında.



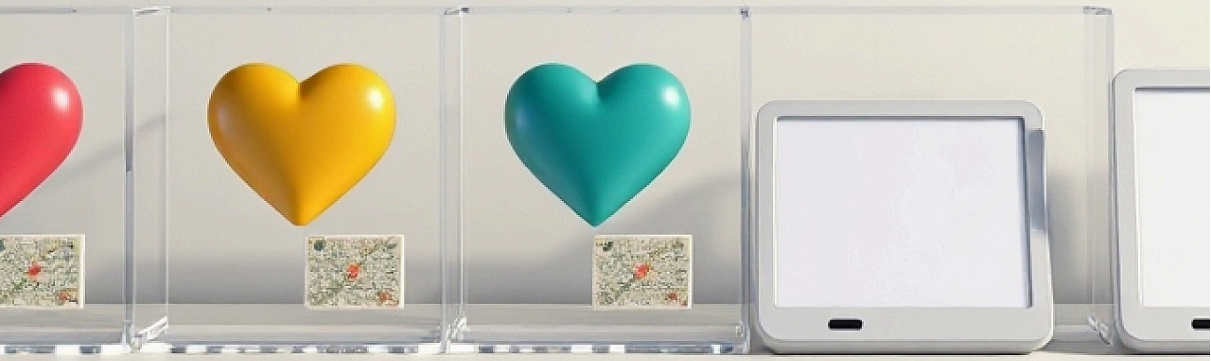
Örnek olarak bir oyundaki karakteri ele alalım. Bu karakterin sağlığı vardır; bu, ne kadar hasar alabileceğini gösteren bir sayıdır. Bir konumu vardır; bu, oyun haritasında nerede durduğunu söyler. Ayrıca bir hareket davranışı, yani konumunu değiştiren bir eylemi vardır. Bu üç şey birbirine aittir ve sonraki sayfalarda bunların birlikte gruplandığını göreceksin.



Karakterin sağlığını, konumunu ve hareket davranışını tek bir yerde birlikte tut. Sağlık veridir, konum veridir ve hareket ikisini de kullanan bir eylemdir. Veri ve eylemlerden oluşan bu demete nesne diyoruz. Bu parçaları programın her yerine dağıtmak yerine nesne onları tek bir birim olarak tutar. Karakter nesnesi bu fikrin somut resmidir.



Bir nesne kendi verisini ve o veri üzerinde çalışan işlemleri tutar. Karakter nesnesi sağlığını ve konumunu saklar; ayrıca konumu okuyup değiştiren hareket eylemini de içerir. Veriye ait eylemi başka yerde aramana gerek yoktur. Nesne onları birlikte tutar, böylece nereye bakacağını her zaman bilirsin.



Oyunda birkaç karakter düşün. Her birinin kendi sağlık değeri ve haritadaki kendi konumu vardır. Yine de hepsi aynı tür yapıyı paylaşır: her birinin sağlığı, konumu ve bir hareket eylemi vardır. Ayrı nesnelerdir, ama aynı kalıbı izlerler. Bu ortak kalıbı bir sınıf tanımlar; bunu birazdan göreceksin.



Bir sınıf, nesnelerin ortak yapısını tanımlar. Nesnelerin hangi verilere sahip olduğunu (sağlık ve konum gibi) ve hangi eylemleri yapabildiklerini (hareket gibi) listeler. Sınıf kalıptır ya da taslaktır; nesneler ondan yapılır. Sınıfı bir kez yazarsın ve her karakter nesnesinin izleyeceği biçimi tanımlar. Bu, tanımı tek bir yerde tutar.



Her karakter nesnesi sınıftan oluşturulur ve kendi değerlerini alır. Bir nesnenin sağlığı 100 ve konumu 3 olabilir, bir başkasının sağlığı 50 ve konumu 7. Sınıftan gelen aynı yapıyı paylaşırlar ama farklı veriler tutarlar. Bir sınıf, birçok nesne; her biri kendi sağlığına ve konumuna sahip. Bir sınıf, her karakter için aynı yapıyı tekrarlamaktan seni böyle kurtarır.



Bir karakteri anlamak için tek bir yere bakarsın: onun nesnesine. Verisi ve eylemleri programın her yerine dağılmamıştır. Hareketin nasıl çalıştığını bilmek istiyorsan hareket eylemini içeren nesneye bakarsın. Bir karakterin sağlığını bilmek istiyorsan aynı nesneye bakarsın. Bakılacak tek bir yerin olması programı okumayı ve değiştirmeyi çok daha kolaylaştırır.



Büyük bir program, her biri kendi nesneleri etrafında kurulmuş anlaşılır parçalara bölünebilir. Tek bir devasa veri ve eylem listesi yerine bir karakter nesnen, bir silah nesnen, bir harita nesnen vb. olur. Her parçanın net bir işi vardır. Amaç, karmaşık programları tek bir düğüm yığını olarak değil, parça parça kavramayı kolaylaştırmaktır.



Anlaşılır parçalar birlikte çalışarak bütün programı oluşturur. Her nesne kendi işini yürütür ve parçalar basit eylemlerle birbirine bağlanır. Karakter nesnesi hareket eder, harita nesnesi konumu gösterir ve silah nesnesi hasar verir. Hiçbir parçanın diğerleri hakkında her şeyi bilmesi gerekmez. Küçük ve net parçaların bu şekilde birleşmesi, nesne yönelimli programlamanın eksiksiz bir programı nasıl kurduğudur.



Nesne yönelimli programlama zorunlu değildir ve her sorun için en iyi yöntem değildir. Bazı sorunlar başka yaklaşımlarla daha basit şekilde ele alınır. Sayıları toplayan küçük bir betik için nesneler oluşturmak gereksiz karmaşıklık ekleyebilir. Bu yaklaşım bir kural değil, bir araçtır. Onu, verileri ve eylemleri nesneler etrafında gruplamak programı netleştirdiğinde seçersin; sırf var olduğu için değil.



Nesne yönelimli programlamanın yanı sıra farklı programlama yaklaşımları da vardır. Bazıları bir programı adım adım bir talimat listesi olarak düzenler. Bazıları veriyi dönüştüren işlevlere odaklanır. Bunu bilmek fikri doğru bağlamda tutar: nesne yönelimli programlama birkaç yoldan biridir. Her yaklaşımın güçlü yanları vardır ve doğru seçim çözdüğün soruna ve ihtiyaç duyduğun netliğe bağlıdır.



Yeni başlayanların sık yaptığı bir hata, her şeyin bir nesneye dönüşmesi gerektiğini varsaymaktır. Kitap, bu yaklaşımın bir kural değil bir seçim olduğunu hatırlatır. Her küçük veri parçasını bir nesneye dönüştürmek programı kolaylaştırmak yerine okumayı zorlaştırabilir. Nesneleri, verileri ve eylemleri gruplamak gerçekten yardımcı olduğunda kullan. Nesnenin programı netleştirip netleştirmediğini sor; netleştirmiyorsa daha basit bir yaklaşım daha iyi olabilir.



Nesne yönelimli programlama, verileri ve eylemleri nesneler etrafında düzenler. Bir sınıf, bu nesnelerin ortak yapısını tanımlar ve bu, karmaşık programları anlaşılır parçalara bölmeye yardımcı olur. Karakter örneği sağlık, konum ve hareketin birlikte tutulduğunu gösterdi. Bunun diğerleri arasında bir yaklaşım olduğunu, bir zorunluluk olmadığını unutma. Programını netleştirdiğinde kullan ve başka yollara açık kal.